FORM PTO-1390 (Rev. 12-29-99) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE **TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371**		ATTORNEY'S DOCKET NUMBER 032326-139 U.S. APPLICATION NO. (If known, see 37 C.F.R. 1.5) Unassigned 09/831745
INTERNATIONAL APPLICATION NO. PCT/FR99/02678	INTERNATIONAL FILING DATE 3 November 1999	PRIORITY DATE CLAIMED 13 November 1998

TITLE OF INVENTION
METHOD AND DEVICE FOR CONTROLLING A PORTABLE OBJECT LIFE CYCLE, IN PARTICULAR, A SMART CARD

APPLICANT(S) FOR DO/EO/US
Marc BIRKNER, Jean Luc GIRAUD and Luarent TALVARD

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and the PCT Articles 22 and 39(1).
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. ☐ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☒ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US)
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
8. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☐ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. ☒ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11. to 16. below concern other document(s) or information included:

11. ☒ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
12. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. ☒ A FIRST preliminary amendment.
    ☐ A SECOND or SUBSEQUENT preliminary amendment.
14. ☐ A substitute specification.
15. ☐ A change of power of attorney and/or address letter.
16. ☐ Other items or information:

JC18 Rec'd PCT/PTO 14 MAY 2001

U.S. APPLICATION NO. (If known, see 37 C.F.R. 1.50) Unassigned 09/831745	INTERNATIONAL APPLICATION NO. PCT/FR99/02678	ATTORNEY'S DOCKET NUMBER 032326-139

17. ☒ The following fees are submitted:				CALCULATIONS	PTO USE ONLY
Basic National Fee (37 CFR 1.492(a)(1)-(5)): Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO .... $1,000.00 (960) International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO .... $860.00 (970) International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO .... $710.00 (958) International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) .... $690.00 (956) International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4) $100.00 (962)					
ENTER APPROPRIATE BASIC FEE AMOUNT =				$ 860.00	
Surcharge of **$130.00 (154)** for furnishing the oath or declaration later than 20 □ 30 □ months from the earliest claimed priority date (37 CFR 1.492(e)).				$ -0-	
Claims	Number Filed	Number Extra	Rate		
Total Claims	36 -20 =	16	X$18.00 (966)	$ 288.00	
Independent Claims	4 -3 =	1	X$80.00 (964)	$ 80.00	
Multiple dependent claim(s) (if applicable)			+ $270.00 (968)	$ -0-	
TOTAL OF ABOVE CALCULATIONS =				$ 1,228.00	
Reduction for 1/2 for filing by small entity, if applicable (see below).				$ -0-	-
SUBTOTAL =				$ 1,228.00	
Processing fee of **$130.00 (156)** for furnishing the English translation later than 20 □ 30 □ months from the earliest claimed priority date (37 CFR 1.492(f)). +				$ -0-	
TOTAL NATIONAL FEE =				$ -0-	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). **$40.00 (581)** per property +				$ -0-	
TOTAL FEES ENCLOSED =				$ 1,228.00	
				Amount to be: refunded	$
				charged	$

a. □ Small entity status is hereby claimed.

b. ☒ A check in the amount of $ 1,228.00 to cover the above fees is enclosed.

c. □ Please charge my Deposit Account No. 02-4800 in the amount of $______ to cover the above fees. A duplicate copy of this sheet is enclosed.

d. □ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 02-4800. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

James A. LaBarre
BURNS, DOANE, SWECKER & MATHIS, L.L.P.
P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

James LaBarre
SIGNATURE

James A. LaBarre
NAME

28,632
REGISTRATION NUMBER

Patent
Attorney's Docket No. 032326-139

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

Marc BIRKNER et al

Application No.: Unassigned

Filed: May 14, 2001

For: METHOD AND DEVICE FOR CONTROLLING A PORTABLE OBJECT LIFE CYCLE, IN PARTICULAR A SMART CARD

Group Art Unit: Unassigned

Examiner: Unassigned

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Prior to examination and the calculation of filing fees, kindly amend the above-identified application as follows:

ALL REFERENCES ARE TO THE TRANSLATION OF THE ANNEX TO THE INTERNATIONAL PRELIMINARY EXAMINATION REPORT (CAPTIONED "Revised Description and Claims").

IN THE SPECIFICATION:

Page 1, immediately following the title appearing on lines 1 and 2, insert the following:

--This disclosure is based upon, and claims priority from French Application No. 98/14517, filed on November 13, 1998 and International Application No. PCT/FR99/02678, filed November 3, 1999, which was published on May 25, 2000 in a language other than English, the contents of which are incorporated herein by reference.

Background of the Invention--

Page 6, before line 13, insert the following heading:

--**Summary of the Invention** --.

Page 8, between lines 12 and 13, insert the following heading:

--**Brief Description of the Drawings**--.

Page 9, between lines 3 and 4, insert the following heading:

--**Detailed Description**--.

Add the following Abstract:

--The invention concerns a device and a method for controlling a portable object life cycle, in particular a smart card, the life cycle being determined by successive state transitions, which states determine the services offered by the object. The object includes a processing unit, program storage units and data storage units, each storage unit having a content defining a plurality of configurations. The device controls the transition from a first state to a second state of the object and, preferably triggers actions when the transition crossover from one state to another occurs or when a transition crossover request is denied. The actions are dependent on the type of transitions implied in the requests for state transition crossover applied to the object.--

IN THE CLAIMS:

Kindly replace claims 1-4, 6-8, 10-14 and 16-36, as follows.

1. (Amended) A device for controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, said states determining the services offered by the object, said object comprising a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), each of said memories (3, 4, 5) having a content defining a plurality of configurations, wherein said device comprises means for controlling the transition from a first state to a second state of the portable electronic object, including means for selectively enabling and/or inhibiting state transitions, so that only some transitions are permitted amongst all the possible transitions.

2. (Amended) A device according to Claim 1, characterised in that the control means comprise means of checking the content of the volatile memory (3), the data

memories (5) and the program memories (4) of the portable electronic object as a function of the state transition to be effected.

3. (Amended) A device according to claim 1, characterised in that the control means enable and/or inhibit a state transition, using a table (11) of permitted state transitions.

4. (Amended) A device according to Claim 3, characterised in that the control means comprise:

- in addition to the table (11) of permitted state transitions;

- a table (12) of the checks to be made per permitted state transition;

- and a check engine (9) using said tables.

6. (Amended) A device according to Claim 4, characterised in that the means of controlling the transition from a first state to a second state of the portable electronic object comprise:

- an extension (16) to the table (11) of permitted state transitions;

- an extension (17) to the table (12) of checks to be made per permitted state transition;

and in that the check engine (9) uses said table extensions (16, 17).

7. (Amended) A device according to claim 1, characterised in that the control means comprise means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object.

8. (Amended) A device according to Claim 7 wherein said controlling means includes:

- an extension (16) to the table (11) of permitted state transitions;

- an extension (17) to the table (12) of checks to be made per permitted state transition;

and in that the check engine (9) uses said table extensions (16, 17); and

wherein said means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object comprise a table of actions (13) which can be used by the check engine (9).

10. (Amended) A portable electronic object having a processing unit (2), a volatile memory (3), program memories (4), data memories (5), and a device for controlling the life cycle of the object comprising means for controlling the transition from a first state to a second state of the portable electronic object, including means for selectively enabling and/or inhibiting state transitions, so that only some transitions are permitted amongst all the possible transitions.

11. (Amended) A smart card having a processing unit (2), a volatile memory (3), program memories (4), data memories (5), and a device for controlling the life cycle of the object comprising means for controlling the transition from a first state to a second state of the portable electronic object, including means for selectively enabling and/or inhibiting state transitions, so that only some transitions are permitted amongst all the possible transitions.

12. (Amended) A method of controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, said states determining the services offered by the object, said object comprising a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), each of said memories (3, 4, 5) having a content defining a plurality of configurations, said method being implemented, within the object, following a state transition request, according to the following steps:

- a step (51, 511, 528, 521) of validation of the enabling of said request using means of enabling and/or inhibiting state transitions, so that only certain transitions are permitted amongst all the possible transitions; and

- a step (57, 517, 527) of modifying the current state of the object if the requested transition is enabled (51, 511, 528, 521).

13. (Amended) A method according to Claim 12, characterised in that it comprises a step (53, 513, 523) of executing systematic actions associated with the requested transition.

14. (Amended) A method according to claim 12 including the following steps

- a step (52, 512, 522) of evaluating checks on the configuration of the object, checks associated with a permitted transition;

- and in that the step (57, 517, 527) of modifying the current state of the object is performed if said checks on the configuration of the object are satisfied (54, 514, 524).

16. (Amended) A method according to claim 14, further including a step (55, 515, 525) of executing negative actions if the checks associated with the requested transition are not satisfied (54, 514, 524).

17. (Amended) A method according to claim 12, further including a step (56, 516, 526) of executing positive actions performed if the requested transition is permitted (51, 511, 528, 521).

18. (Amended) A method according to claim 12, implemented within the object, following a request for transition from a first reference state to a second reference state, characterised in that the step (51) of validating the enabling of the said request consists of analysing a table (11) of permitted transitions.

19. (Amended) A method according to Claim 18, including the steps of:

- using an entry (400, 401), corresponding to the requested transition, in a table (13) of actions, and

- executing a programme of actions (404) defined by the said entry.

20. (Amended) A method according to claim 18, further including the step (52) of evaluating the checks associated with the requested transition comprising the steps of:

- using an entry (30) in a table (12) of checks, and

- executing a programme (32) of checks defined by said entry.

21. (Amended) A method according to claim 18 further including the step (56) of executing positive actions, if the requested transition is enabled (51) and if the checks associated with the requested transition are satisfied (54), comprising the steps of:

- using an entry (400, 402), corresponding to the requested transition, in a table (13) of actions, and

- executing a programme (405) of actions defined by the said entry.

22. (Amended) A method according to claim 18 further including the step (55) of executing negative actions if the checks associated with the requested transition are not satisfied (54), comprising the steps of:

- using an entry (400, 403), corresponding to the requested transition, in the table (13) of actions, and

- executing a programme (406) of actions defined by the said entry.

23. (Amended) A method according to claim 18, further including the step (56) of executing positive actions if the requested transition is enabled (51), comprising the steps of:

- using an entry (400, 402), corresponding to the requested transition, in the table (13) of actions, and

- executing a programme (405) of actions defined by the said entry.

24. (Amended) A method according to claim 12, implemented within the object, following a request for transition from a first additive state to a second additive state, characterised in that the step (511) of validating the enabling of said request consists of analysing an extension (16) to a table (11) of permitted transitions.

25. (Amended) A method according to Claim 24, further including the step (513) of executing systematic actions comprising the steps of:

- using an entry (407, 408), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme of actions (411) defined by said entry.

26. (Amended) A method according to claim 24, further including the step (512) of evaluating the checks associated with the requested transition, comprising the steps of:

- using an entry (33) in an extension (17) to a table (12) of checks, and

- executing a programme (35) of checks defined by said entry.

27. (Amended) A method according to claim 24 further including the step (516) of executing positive actions if the requested transition is enabled (511) and if the checks associated with the requested transition are satisfied (514), comprising the steps of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (412) of actions defined by said entry.

28. (Amended) A method according to claim 24, further including the step (515) of executing negative actions if the checks associated with the requested transition are not satisfied (514), comprising the steps of:

- using an entry (407, 410), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (413) of actions defined by [the] said entry.

29. (Amended) A method according to claim 24, further including the step (516) of executing positive actions if the requested transition is enabled (511), of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to the table (13) of actions, and

- executing a programme (412) of actions defined by said entry.

30. (Amended) A method according to claim 12, implemented within the object, following a request for transition from a reference state to an additive state, characterised in that the step (528, 521) of validating the enabling of said request consists of:

- validating (528) the enabling of a transition from said reference state to an additive state, whilst analysing a table (11) of permitted transitions;

- validating (521) the enabling of a transition from said reference state to said additive state, whilst analysing an extension (16) to the table (11) of permitted transitions.

31. (Amended) A method according to Claim 30, further including the step (513) of executing systematic actions comprising the steps of:

- using an entry (407, 408) corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme of actions (411) defined by said entry.

32. (Amended) A method according to claim 30, further including the step (522) of evaluating the checks associated with the requested transition comprising the steps of:

- using an entry (33) in an extension (17) to a table (12) of checks, and

- executing a programme (35) of checks defined by said entry.

33. (Amended) A method according to claim 30, further including the step (526) of executing positive actions if the requested transition is enabled (528, 521) and if the checks associated with the requested transition are satisfied (524), comprising the steps of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (412) of actions defined by said entry.

34. (Amended) A method according to claim 30, further including the step (525) of executing negative actions if the checks associated with the requested transition are not satisfied (524), comprising the steps of:

- using an entry (407, 410), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (413) of actions defined by said entry.

35. (Amended) A method according to claim 30, further including the step (526) of executing positive actions if the requested transition is enabled (528, 521), of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to the table (13) of actions, and

- executing a program (412) of actions defined by said entry.

36. (Amended) A method according to claim 12, wherein said method does not enable the crossover of a state transition, from an additive state to a reference state.

REMARKS

Entry of the foregoing amendment is respectfully requested. This amendment is intended to eliminate the multiple dependency of the claims.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: James LaBarre
James A. LaBarre
Registration No. 28,632

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

Date: May 14, 2001

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

1. (Amended) A device for controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, [the] said states determining the services offered by the object, [the] said object comprising a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), each of [these] said memories (3, 4, 5) having a content defining a plurality of configurations, [characterised in that it has means of] wherein said device comprises means for controlling the transition from a first state to a second state of the portable electronic object, [using means of] including means for selectively enabling and/or inhibiting state transitions, so that only some transitions are permitted amongst all the possible transitions.

2. (Amended) A device according to Claim 1, characterised in that the control means comprise means of checking the content of the volatile memory (3), the data memories (5) and the program memories (4) of the portable electronic object as a function of the state transition [permitted] to be effected.

3. (Amended) A device according to [either one of Claims 1 or 2] claim 1, characterised in that the control means enable and/or inhibit a state transition, using a table (11) of permitted state transitions.

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

4. (Amended) A device according to Claim 3, characterised in that the control means comprise:

- in addition to the table (11) of permitted state transitions;

- a table (12) of the checks to be made per permitted state transition;

- and a check engine (9) using [the] said tables.

6. (Amended) A device according to Claim 4, characterised in that the means of controlling the transition from a first state to a second state of the portable electronic object comprise:

- an extension (16) to the table (11) of permitted state transitions;

- an extension (17) to the table (12) of checks to be made per permitted state transition;

and in that the check engine (9) uses [the] said table extensions (16, 17).

7. (Amended) A device according to [any one of Claims 1 to 6] claim 1, characterised in that the control means comprise means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object.

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

8. (Amended) A device according to Claim 7 [when the latter depends on Claims 5 or 6, characterised in that the] wherein said controlling means includes:

- an extension (16) to the table (11) of permitted state transitions;

- an extension (17) to the table (12) of checks to be made per permitted state transition;

and in that the check engine (9) uses [the] said table extensions (16, 17); and

wherein said means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object comprise a table of actions (13) which can be used by the check engine (9).

10. (Amended) A portable electronic object having a processing unit (2), a volatile memory (3), program memories (4), [and] data memories (5), [characterised in that it includes the] and a device for controlling the life cycle of the object[, according to one of Claims 1 to 9] comprising means for controlling the transition from a first state to a second state of the portable electronic object, including means for selectively enabling and/or inhibiting state transitions, so that only some transitions are permitted amongst all the possible transitions.

11. (Amended) A smart card having a processing unit (2), a volatile memory (3), program memories (4), [and] data memories (5), [characterised in that it includes the]

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

and a device for controlling the life cycle of the object[, according to one of Claims 1 to 9] comprising means for controlling the transition from a first state to a second state of the portable electronic object, including means for selectively enabling and/or inhibiting state transitions, so that only some transitions are permitted amongst all the possible transitions.

12. (Amended) A method of controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, [the] said states determining the services offered by the object, [the] said object comprising a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), each of [these] said memories (3, 4, 5) having a content defining a plurality of configurations, [the] said method being implemented, within the object, following a state transition request, according to the following steps:

[characterised in that it comprises:]

- a step (51, 511, 528, 521) of validation of the enabling of [the] said request using means of enabling and/or inhibiting state transitions, so that only certain transitions are permitted amongst all the possible transitions; and

- a step (57, 517, 527) of modifying the current state of the object if the requested transition is enabled (51, 511, 528, 521).

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

13. (Amended) A method according to Claim 12, characterised in that it comprises a step (53, 513, 523) of executing systematic actions[, actions] associated with the requested transition.

14. (Amended) A method according to [either one of Claims 12 or 13, characterised in that it comprises:] claim 12 including the following steps

- a step (52, 512, 522) of evaluating checks on the configuration of the object, checks associated with a permitted transition;

- and in that the step (57, 517, 527) of modifying the current state of the object is performed if [the] said checks on the configuration of the object are satisfied (54, 514, 524).

16. (Amended) A method according to [either one of Claims 14 or 15, characterised in that it comprises] claim 14, further including a step (55, 515, 525) of executing negative actions if the checks associated with the requested transition are not satisfied (54, 514, 524).

17. (Amended) A method according to [either one of Claims 12 to 13, characterised in that it comprises] claim 12, further including a step (56, 516, 526) of

executing positive actions performed if the requested transition is permitted (51, 511, 528, 521).

18. (Amended) A method according to [any one of Claims 12 to 17] claim 12, implemented within the object, following a request for transition from a first reference state to a second reference state, characterised in that the step (51) of validating the enabling of the said request consists of analysing a table (11) of permitted transitions.

19. (Amended) A method according to Claim 18, [when the latter depends on Claims 13 to 17, characterised in that the step (53) of executing systematic actions consists] including the steps of:

- using an entry (400, 401), corresponding to the requested transition, in a table (13) of actions, and
- executing a programme of actions (404) defined by the said entry.

20. (Amended) A method according to [either one of Claims 18 or 19, when Claim 18 depends on Claims 14 to 16, characterised in that] claim 18, further including the step (52) of evaluating the checks associated with the requested transition [consists] comprising the steps of:

- using an entry (30) in a table (12) of checks, and

- executing a programme (32) of checks defined by [the] said entry.

21. (Amended) A method according to [any one of Claims 18 to 20, when Claim 18 depends on Claims 15 to 16, characterised in that] claim 18 further including the step (56) of executing positive actions [consists], if the requested transition is enabled (51) and if the checks associated with the requested transition are satisfied (54), comprising the steps of:

- using an entry (400, 402), corresponding to the requested transition, in a table (13) of actions, and

- executing a programme (405) of actions defined by the said entry.

22. (Amended) A method according to [any one of Claims 18 to 21, when Claim 18 depends on Claim 16, characterised in that] claim 18 further including the step (55) of executing negative actions [consists,] if the checks associated with the requested transition are not satisfied (54), comprising the steps of:

- using an entry (400, 403), corresponding to the requested transition, in the table (13) of actions, and

- executing a programme (406) of actions defined by the said entry.

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

23. (Amended) A method according to [either one of Claims 18 to 19, when Claim 18 depends on Claim 17, characterised in that] claim 18, further including the step (56) of executing positive actions [consists,] if the requested transition is enabled (51), comprising the steps of:

- using an entry (400, 402), corresponding to the requested transition, in the table (13) of actions, and

- executing a programme (405) of actions defined by the said entry.

24. (Amended) A method according to [any one of Claims 12 to 17] claim 12, implemented within the object, following a request for transition from a first additive state to a second additive state, characterised in that the step (511) of validating the enabling of [the] said request consists of analysing an extension (16) to a table (11) of permitted transitions.

25. (Amended) A method according to Claim 24, [when the latter depends on Claims 13 to 17, characterised in that] further including the step (513) of executing systematic actions [consists] comprising the steps of:

- using an entry (407, 408), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme of actions (411) defined by [the] said entry.

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

26. (Amended) A method according to [either one of Claims 24 or 25, when Claim 24 depends on Claims 14 to 16, characterised in that] claim 24, further including the step (512) of evaluating the checks associated with the requested transition, comprising the steps [consists] of:

- using an entry (33) in an extension (17) to a table (12) of checks, and
- executing a programme (35) of checks defined by [the] said entry.

27. (Amended) A method according to [any one of Claims 24 to 26, when Claim 24 depends on Claims 15 to 16, characterised in that] claim 24 further including the step (516) of executing positive actions [consists,] if the requested transition is enabled (511) and if the checks associated with the requested transition are satisfied (514), comprising the steps of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and
- executing a programme (412) of actions defined by [the] said entry.

28. (Amended) A method according to [any one of Claims 24 to 27, when Claim 24 depends on Claim 16, characterised in that] claim 24, further including the step

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

(515) of executing negative actions [consists,] if the checks associated with the requested transition are not satisfied (514), comprising the steps of:

- using an entry (407, 410), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (413) of actions defined by [the] said entry.

29. (Amended) A method according to [either one of Claims 24 to 25, when Claim 24 depends on Claim 17, characterised in that] claim 24, further including the step (516) of executing positive actions [consists,] if the requested transition is enabled (511), of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to the table (13) of actions, and

- executing a programme (412) of actions defined by [the] said entry.

30. (Amended) A method according to [any one of Claims 12 to 17] claim 12, implemented within the object, following a request for transition from a reference state to an additive state, characterised in that the step (528, 521) of validating the enabling of [the] said request consists of:

- validating (528) the enabling of a transition from [the] said reference state to an additive state, whilst analysing a table (11) of permitted transitions;

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

- validating (521) the enabling of a transition from [the] said reference state to [the] said additive state, whilst analysing an extension (16) to the table (11) of permitted transitions.

31. (Amended) A method according to Claim 30, [when the latter depends on Claims 13 to 17, characterised in that] further including the step (513) of executing systematic actions [consists] comprising the steps of:

- using an entry (407, 408) corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme of actions (411) defined by [the] said entry.

32. (Amended) A method according to [either one of Claims 30 or 31, when Claim 30 depends on Claims 14 to 16, characterised in that] claim 30, further including the step (522) of evaluating the checks associated with the requested transition [consists] comprising the steps of:

- using an entry (33) in an extension (17) to a table (12) of checks, and

- executing a programme (35) of checks defined by [the] said entry.

33. (Amended) A method according to [any one of Claims 30 to 32, when Claim 30 depends on Claims 15 to 16, characterised in that] claim 30, further including the

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

step (526) of executing positive actions [consists,] if the requested transition is enabled (528, 521) and if the checks associated with the requested transition are satisfied (524), comprising the steps of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (412) of actions defined by [the] said entry.

34. (Amended) A method according to [any one of Claims 30 to 33, when Claim 30 depends on Claim 16, characterised in that] claim 30, further including the step (525) of executing negative actions [consists,] if the checks associated with the requested transition are not satisfied (524), comprising the steps of:

- using an entry (407, 410), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (413) of actions defined by [the] said entry.

35. (Amended) A method according to [either one of Claims 30 to 31, when Claim 30 depends on Claim 17, characterised in that] claim 30, further including the step (526) of executing positive actions [consists,] if the requested transition is enabled (528, 521), of:

Attachment to Preliminary Amendment dated May 14, 2001

Marked-up Claims 1-4, 6-8, 10-14, 16-36

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to the table (13) of actions, and

- executing a program (412) of actions defined by [the] said entry.

36. (Amended) A method according to [any one of Claims 12 to 35, characterised in that the] claim 12, wherein said method does not enable the crossover of a state transition, from an additive state to a reference state.

5/PRTS

Revised description and claims

A METHOD AND DEVICE FOR CONTROLLING THE LIFE CYCLE OF A PORTABLE OBJECT, NOTABLY A SMART CARD

The invention concerns portable electronic objects such as electronic microcircuit cards, known as smart cards, which, connected to electronic devices to enable the latter to perform particular functions in the context of one or more applications, require their life stages to be controlled. The said cards are in fact generally used in applications (banking, communication, identity, health etc) requiring a high degree of security against fraudulent usage. Thus, by way of example, the document US 5473690 presents a smart card comprising several applications, access to which is protected by passwords, a password being dedicated to a user. Knowing a password, it is possible to select one application or another. However, it is not possible to deactivate an application or limit the use thereof whatever the user of the card as a function of the life stages of the said card.

The invention applies more generally to any independent on-board system provided with a processing unit and program and data memories.

In the world of smart cards it is known that the latter result from assembling a component (generally comprising a microprocessor in relationship with memories via communication buses), a module (produced by means of a conductive metal) to which the said component is connected (in the context of a so-called contact smart card) to enable the said component to be connected to an electronic reading and/or writing device (or coupler) and a card body or more generally a support on which the module/component assembly is integrated. In the context of a so-called contactless smart card, the said module is replaced by an antenna and the assembly formed by the component and the said antenna is integrated within the said support.

The life of a smart card can generally be broken down into two sets of stages following each other, corresponding respectively to the manufacture and use of the said card. Putting together the two sets of stages forms a life cycle of the said card. The manufacture of a smart card (with or without contact) consists of several stages.

This is because it is first of all necessary to have an electronic component which is initialised, insulated and then connected to a module. The said component and the module to which it is connected are subsequently integrated on or within a support (generally a plastic card body) itself printed for the purpose of identification or advertising. Subsequently the smart card thus obtained is initialised or programmed in order to meet the conditions of use in the context of applications.

The second set of life stages of a smart card corresponds to its use. This set can itself be divided into several stages, each corresponding, for example, to the implantation or elimination of services offered by the smart card to the user according to his profile, for example.

In addition different participants (component manufacturer, smart card manufacturer, card personalisation centre, card issuer or card carrier) act during the different stages of manufacture and use of a smart card. Thus the components are supplied and sometimes partly initialised by electronic component manufacturers on a silicon wafer. This phase corresponds to the step of manufacturing the component. The following step is the embedding phase carried out by the smart card manufacturer. It includes the insulation of a component from the silicon wafer, the connection of the said component to a module (or antenna), and the integration of the assembly on the support or card body. There follows the preparation of the application structure present in the electrically programmable memory of the component. This is the electrical personalisation stage which is carried out by the manufacturer of the smart cards or by a personalisation centre or a third party specialising in personalisation of cards or by the issuer himself who is ultimately responsible for the distribution of the cards on the market. This electrical personalisation phase can therefore be broken down into as many stages as there are players or intermediaries. Subsequently, during the use of the smart card, we have seen previously that it can be advantageous to distinguish several stages along with the change in the profile of the card user for example. For all these reasons, it is therefore important to rigorously monitor the life stages of a card in order to know at any time the current stage of the said card within its life cycle. In addition, it is essential on the one hand for access to the electrically programmable memory of a card component in write or read mode to be protected during the exchange of the said card (or component) during the different players and on the other hand for access to the said memory to be limited as the life stages of the card mentioned above follow each other, by activating or deactivating services for example. Finally, it is also sometimes necessary to validate the application context of the smart card before the carrier thereof uses it on the market. For example, a person issuing a smart card of the electronic purse type must be certain that the balance of the said card is indeed zero before issuing the card.

In order to attempt to meet these requirements, different solutions are used at the present time. Certain solutions are purely external to the smart card (physical security at the premises where the said card is manufactured, use of transportation means which are themselves made secure etc). Other solutions complementary to the first, but this time internal or implanted in the card, are also generally used. Use is thus made of secrets for protecting access to the component memory in read/write mode and also logic indicators for irreversibly monitoring the different life stages of the card. For this purpose, bits within a non-erasable memory of the component of the smart card are positioned at the active state at the end of the different life stages of the card (manufacture and initialisation of the component by the manufacturer of the said component, embedding and initialisation of the card memory by the smart card manufacturer, preparation of the application structure of the smart card memory by the personalisation centre or the card issuer etc). According to these indicators, the program (or operating system) executed by the microprocessor of the smart card component, implanted within one of the memories of the said card component, adapts its behaviour as the life stages of the said card follow each other. Thus functions can be modified, added or eliminated.

Whatever the solutions used at the present time, they are all based on the fact that the different players involved in the manufacture of a card are trusted third parties. Only persons liable to intercept components or cards during their transfer between two of the different players are deemed to be "potential fraudsters" and the solutions disclosed above make it possible to be free of them. The adaptation of the operating system of the card according to irreversible indicators affords a not insignificant advantage. Thus, if the manufacturers of the components or cards inscribe systems data or secrets, the card issuer will for example not be able to dispense freely with the said secrets or modify the said system data. However, this solution does not resolve the problem of a fraudulent initialisation of the card or an inopportune error during the said initialisation, carried out by one of the participants.

The invention proposes to remedy the drawbacks of the current state of the art. In particular, the invention consists of providing the operating system of a smart card with software means enabling the said operating system to control an irreversible change in life stage of the said card according to a set of checks on the content of the memories of this same smart card. In addition the invention makes provision, during a change in life stage, for the operating system of the card to be able to automatically trigger actions for adapting the services offered by the said operating system of the said card.

To this end, the invention concerns a device for controlling the life cycle of a portable electronic object, the life cycle consisting of a succession of state transitions, the said states determining the services offered by the object, the said object comprising a processing unit, a volatile memory, program memories and data memories, each of these memories having a content defining a plurality of configurations, characterised in that it has means of controlling the transition from a first state to a second state of the portable electronic object.

According to other characteristics of the device according to the invention:

- the control means have:

- means of enabling and/or inhibiting state transitions to be effected;

- means of checking the content of the volatile memory, of the data memories and of the program memories of the portable electronic object according to the state transition to be effected;

- means for triggering actions during the processing of a request to effect a state transition.

In addition, the invention concerns a portable electronic object, which may notably be a smart card, containing the said life cycle control device.

Moreover, the invention concerns a method of controlling the life cycle of a portable electronic object, the said method being implemented within the object following a state transition request,

characterised in that it comprises:

- a step of validating the enabling of the said request;

- a step of evaluating the checks associated with the requested transition;

- a step of modifying the current state of the object if and only if the requested transition is enabled and if the checks on the configuration of the object are satisfied.

According to other characteristics, the method possibly also comprises:

- a step of executing systematic actions;

- a step of executing positive actions in the case where the requested transition is enabled and if the checks associated with the requested transition are satisfied;

- a step of executing negative actions in the case where the checks associated with the requested transition are not satisfied.

The invention will be understood more clearly from a reading of the following description and an examination of the figures which accompany it. These are given only as an indication and are in no way limitative of the invention.

The figures show:

- Figure 1: a component of a smart card provided with a state transition check device;

- Figures 2a and 2b: a detailed representation of a state transition table;

- Figure 3: a detailed representation of a transition check table;

- Figure 4: a detailed representation of an action table;

- Figure 5: a description of the steps implemented in the method used by the transition check device;

- Figures 6a to 6d: the particularities implemented in the case of an example of a smart card of the electronic purse type.

In the invention, the term reference state will refer to a state from which it is possible to switch to another state following the crossover of a transition described in the table of transitions, located in the program memory. As described below, it is possible to add new states and therefore new transitions after the step of manufacturing the component has taken place. In this case, additive states will be spoken of in order to characterise these in contradistinction to reference states. In addition, the state in which the on-board system is will be referred to as the current state.

Figure 1 shows a component 1, of a smart card, provided with a transition check device according to the invention. The component has a processing unit 2 or a microprocessor in relationship with memories 3, 4 and 5 via a communication bus 6. A non-erasable program memory 4 (or a ROM) has on the one hand a program area 7, the said programs (or on-board system) being able to be executed by the said processing unit and on the other hand a predefined data area 10 which contains constants used by the said operating system. Amongst the said constants of the area 10, the operating system 7, containing a program referred to as a check engine 9, uses a table of transitions 11 which makes it possible to specify the states to which it is possible to gain access from the current state, a check table 12 which makes it possible to associate with each state transition checks relating to the content of the memories 3, 4 and/or 5. In a variant, the check engine 9 can automatically trigger actions when a transition is crossed over or this crossover is rejected. For this purpose, the area 10 of the program memory contains a table of actions 13 which makes it possible to associate actions to be performed with each possible state transition.

A volatile memory 3 (or RAM, standing for Random Access Memory in English) enables the processing unit 2 to temporarily store results or secrets issuing from calculations described by the programs implanted in the program memory 4. The content of the memory 3 is erased each time the component 1 is powered up or each time resetting thereof is requested.

A data memory 5, electrically erasable, generally using EEPROM technology (standing for Electrically Erasable Programmable Read Only Memory in English) has an area 14 containing the variable data necessary for executing the programs 7. This area 14 contains notably a data item 8 referred to as the "current state" making it possible to store the current state of the portable electronic object. The data memory 5 also has an area 15 comprising optionally extensions to the tables 11 to 13 in the case where it is necessary to add states to the reference states. The area 15 then contains an extension to the table of transitions 16 and an extension to the check table 17 and may include an extension to the table of actions 18 if it is wished to associate actions with the new additive state transitions, as seen previously with regard to table 13. In the case of adding states with respect to the reference states, it is sometimes essential to enhance the operating system 7. For this purpose, the memory 5 can also include an area 19 which contains the additional programs which will be executed in their turn by the processing unit 2.

Figure 2a shows a possible use of the table of transitions 11. If it is assumed that i reference states are counted, it is possible to imagine a transition table comprising i columns and i rows. The columns correspond to the reference states which, at a given time, can be the current state. The first i rows correspond to the reference states to which access can be gained from the current state. Thus the value of a box in the table of transitions 11 corresponding to the intersection of a row and column in the said table makes it possible to code either the absence of an enabled transition (zero value for example - this is the case with the transition 20) or the enabling of a transition (non-zero value - this is the case with the transition 21). In the case of an enabled transition, the transition check engine searches within the check table 12 the checks to be made in order to accept or reject the crossover of the requested transition.

Figure 2b also shows a possible implementation of a transition table in the case where it is possible to add states (additive states) to the reference states. The table of transitions includes an additional line compared with Figure 2a. The (i+1)th line makes it possible to specify if transitions from a current reference state to an additive state are enabled. Thus the value of the box 22 indicates an inhibited transition from a reference state to an additive state. The box 23 indicates that it will be possible to switch from the reference state Ei to an additive state. An extension 16 to the table of transitions is then necessary. The latter has j lines corresponding to j additive states to which it is possible to gain access from the (i+j) possible current states represented by the (i+j) columns of the extension 16 to the table of transitions. Thus the combination of the box 23 in the table of transitions and the box 24 of the extension 16 of the table of transitions indicates to the check engine that it is possible to switch from the reference state Ei to the additive state E(i+1).

Figure 3 shows a use of the check table. The check table 12 is located within the area 10 of the predefined data of the memory 4. Each enabled transition has an entry in the said table. An entry comprises a field 30 for identifying the transition and a field 31 containing a reference (or address) to a program 32 of the operating system 7. The check engine 9 can thus make the processing unit 2 execute the required controls for accepting the crossover of the transition. Figure 3 also illustrates a structure of an extension 17 to the check table. In the same way as with the table 12, the extension to the check table 17 has one entry per possible transition. Each entry comprises two fields, a field 33 for identifying the transition and a field 14 containing a reference (or address) of a program 35 of the operating system or, as shown by Figure 3, an additional program located in the data memory 5 (in the area 19).

Figure 4 shows a representation of the table of actions 13 located in the area 10 of the predefined data of the program area 4. At the time of a transition crossover request, it is possible to trigger actions. These can be of three types: systematic action, positive action (that is to say dependent on the fact that the checks are satisfactory) or negative action (that is to say dependent on the fact that the checks are not satisfactory). Figure 4 shows that, at each enabled transition, there is an entry in the table of actions 13. This entry comprises four fields. The first field 400 identifies the transition. The other three fields 401, 402 and 403 each contain a reference or address of a program 404, 405 or 406 of the operating system. The field 401 is dedicated to a systematic action, the field 402 to a positive action and the field 403 to a negative action. Figure 4 also shows an extension 18 to the table of actions. This table 18 is located in the area 15 of the data memory 5 of the component 1. In the same way as with the table of actions 13, the extension to the table of actions 18 comprises one entry per possible transition. An entry comprises four fields. The first field 407 identifies the transition. The other three fields 408, 409 and 410 each contain a reference or address of a program

411, 412 or 413 of the operating system or, as shown by Figure 4, programs located in the area 19 of the data memory 5 of the component 1. The field 408 is dedicated to a systematic action, the field 409 to a positive action and the field 410 to a negative action.

Figure 5a describes the method for validating or rejecting the crossover of a state transition, from a first reference state to another reference state. The request for crossover of a transition can be formulated following an instruction from the card manufacturer or by any other player in the life cycle of the smart card. The said request can also be formulated directly by the card itself, for example through an action associated with a transition. In the context of Figure 5a, the current reference state is the state Ei. The instruction 50 to switch from the state Ei to the state Ej is formulated. Step 51 consists of checking, within the table of transitions 11, that the transition from the state Ei to the state Ej is enabled. Where this transition is inhibited, the transition crossover request 50 is rejected. The current state remains the state Ei. On the other hand, if the transition is enabled, the check engine 9 executes the checks associated with the said transition. For this purpose the check engine evaluates the entry in the check table 12 dedicated to the transition T(Ei→Ej). The execution of said checks corresponds to step 52 of the method. The check engine 9 executes the systematic actions associated with the transition T(Ei→Ej) according to the entry in the table of actions 13 dedicated to the said transition (step 53). If the checks 54 required at the time of the request for crossover of the transition 50 are not satisfactory, the current state remains unchanged. According to the entry in the table of actions 13 associated with the transition T(Ei→Ej) the check engine executes the negative actions (step 55 of the method). The performance of the method is then terminated. On the other hand, if the checks 54 are satisfactory, then the current state becomes the state Ej (step 56 of the method). The positive actions are then executed (step 57 of the method) according to the state of the entry in the table of actions 13 associated with the transition T(Ei→Ej). The performance of the method is terminated.

Figure 5b describes the method for validating or rejecting the crossover of a state transition, from a first additive state to another additive state. The current additive state is the state Ei. The instruction 510 to switch from the additive Ei to the additive state (or reference state) Ej is formulated. Step 511 of the method consists of checking within the extension to the table of transitions 16 that the transition from state Ei to state Ej is enabled. Where this transition is inhibited, the transition crossover request 510 is rejected. The current state remains the state Ei. On the other hand, if the transition is enabled, the check engine 9 executes the checks associated with the said transition. For this purpose, the check engine evaluates the entry in the extension to the check table 17 dedicated to the transition

T(Ei→Ej). The execution of the said checks constitutes step 512 of the method. The check engine 9 executes the systematic actions associated with the transition T(Ei→Ej) according to the entry in the extension to the table of actions 18 dedicated to the said transition (step 513 of the method). If the check 514 required at the time of the transition crossover request 510 is not satisfactory, the current state remains unchanged. According to the entry in the extension to the table of actions 18 associated with the transition T(Ei→Ej), the check engine 9 executes the negative actions (step 515 of the method). The performance of the method is then terminated. On the other hand, if the checks 514 are satisfactory, the current state becomes state Ej (step 516 of the method). The positive actions are then executed (step 517 of the method) according to the state of the entry in the extension to the table of actions 18 associated with the transition T(Ei→Ej). The performance of the method is terminated.

Figure 5c describes the method for validating or rejecting the crossover of a state transition, from a reference state to an additive state. The current reference state is the state Ei. The instruction 520 to switch from the reference state Ei to the additive state Ej is formulated. Step 528 of the method consists of checking, within the table of transitions 11, that a transition from the current reference Ei to an additive state is enabled. If such a transition is inhibited, the method is terminated. The current state remains unchanged. On the other hand, if a transition from the said reference state to an additive state is enabled, the check engine runs steps 521 to 527 of the method, respectively identical to steps 511 to 517 described in relation to Figure 5b.

An example of an application in the field of electronic purses is presented in relation to Figures 6a to 6d. The said application makes it possible to pay for purchases by means of "electronic money" stored in a smart card, instead of paying in cash. The use of such a technique requires a management of the cards which is as secure as that which would have been imposed by the use of cash. It is necessary for example to avoid the creation of paper money. The security of an electronic purse smart card is generally based on keys stored within the smart card allowing secure transactions using cryptography. Such a card has an operating system offering a set of commands and services for crediting or debiting money. At the start of the life cycle of the electronic purse smart card, the said smart card is not initialised. It contains no information. Figure 6a shows the predefined reference states:

- State E1 "blank card" (referenced 80): only test commands for validating the behaviour of the data memory 5 are available (verification that the EEPROM technology memory boxes can be correctly written to and erased);

- State E2 "tested card" (referenced 82): the test commands are no longer available. In their turn commands generally known as "physical commands" (allowing access in write mode by means of a physical addressing independently of any logic structure of the file type for example) are available. They make it possible to initialise the card (writing in the area 14 of the data memory of the logic constituents necessary for the functioning of the application, that is to say files, balances etc);

- State E3 "initialised card" (referenced 84): the physical commands are no longer available. Logic commands for personalising the card (addition of new logic structures and initialisation data in the said structures) can be used. In addition, a recovery mechanism is activated so that the smart card does not lose the coherence of these data when it is powered down during the execution of one of the said logic commands;
- State E4 "personalised card" (referenced 86): the logic commands specific to the electronic purse application (debit/credit) are activated.

The set of available commands changes according to the life stage in which the smart card is situated. Information stored in data memory enables the operating system to know the state in which the smart card is situated. Figure 6a also shows that, in the context of a card of the electronic purse type, all the transitions between reference states must be crossed successively (from state E1 to state E4), and this irreversibly. Any other transition is inhibited. Only the possibility of subsequently using additive states 88 is offered. This possible transition is referenced

87. The operating system according to the current state allows only a set of commands specific to each reference state.

The checks and actions to be triggered when a transition is crossed are described as follows:

- Transition from state E1 to state E2 (denoted T(E1→E2) and referenced 81):

- Check: none

- Systematic action:

erasure of the data memory in order to prevent a fraudster leaving therein data which can be interpreted by the card operating system;

- Transition from state E2 to state E3 (denoted T(E2→E3) and referenced 83):

- Check:

- integrity of the data written in the data memory with the physical commands (validation of a redundancy code by data);

- verification of the blank state of the memory apart from the said data;

- Positive action:

- activation of the recovery mechanism;

- Transition from state E3 to state E4 (denoted T(E3→E4) and referenced 85):

- Verification:

- nullity of the balance of the electronic purse

- Action: none

- Transition from state E4 to an additive state (denoted T(E4→Eadd) and referenced 87):

- Verification: none

- Action: none

Figures 6b to 6d illustrate respectively an embodiment of a table of transitions 11, a check table 12 and a table of actions 13, according to the invention. The table of transitions 11 as described in relation to Figure 6b makes it possible to enable only the transitions 81, 83, 85 and 87. For this, only the boxes 60 to 63 in the said table contain a non-zero value. The other boxes in the table of transitions contain a zero value in order to indicate that any other transition is inhibited. The check table as presented through Figure 6c makes it possible to associate the checks to be satisfied for enabling the crossover of the transitions 81, 83, 85 and 87, the said transitions enabled by the table of transitions 11 (Figure 6b). Thus the entry 64 in the check table 12 includes a field 641 for identifying the fact that the said entry is dedicated to the transition 81. The entry 64 also includes a field 642 containing a zero reference in order to indicate that no check is requested in order to allow the crossover of the transition 81. In a variant, the transition 81 has no associated entry. This variant is illustrated later in the case of the table of actions. The check table 12 has an entry 65 which comprises respectively a field 651 for indicating that the entry is associated with the transition 83 and a field 652 containing the reference of a program 67, located in the program memory, so that the check engine can make the checks described above. Likewise, the check table 12 has an entry 65 which comprises respectively a field 661 for indicating that the entry is associated with the transition 83 and a field 662 containing the reference of a program 68, located in the program memory, so that the check engine can make the previously described checks.

Figure 6d presents an embodiment of the table of actions 13. The said table has an entry 71 which includes a field 711 for indicating that the said entry is associated with the transition 81. The same entry 71 has a field 712 containing the reference of a program 75, located in the program memory, so that the check engine can execute the systematic actions associated with the transition 81. The entry 71 also has a field 713 and a field 714 containing a zero reference in order to indicate to the check engine that no positive or negative action is associated with the crossover of the transition 81. In the same way, the table of actions 13 has a second entry 72 comprising the fields 721 to 724 in order to indicate to the check engine that the said entry is associated with the transition 83, that the program 74 is to be executed as a positive action when the said transition is crossed and that no systematic or negative action is to be executed. The absence of entry, within the table of actions 13, associated with the transition 85, indicates that no action (systematic, positive or negative) is to be executed at the time of crossover or rejection of crossover of the said transition.

By means of the device and method as described above, the life cycle of a portable electronic object is controlled. Each state transition is irreversible and the checks made at the time of each transition request guarantee a coherent memory configuration for the object. In addition, the systematic, positive or negative actions make it possible to adapt the behaviour of the said object. Finally, in the case where provision is made for enabling one or more transitions from one or more reference states to an additive state, the life cycle of the object can easily be enhanced, for example after the object is issued on the market, without the predefined life cycle (composed of a succession of transitions from one reference state to another reference state) being able to be diverted.

Any risk of fraud during the initialisation of a portable electronic object or of an inopportune error during the said initialisation is removed whilst preserving great adaptability of control of the life cycle of the object.

CLAIMS

1. A device for controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, the said states determining the services offered by the object, the said object comprising a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), each of these memories (3, 4, 5) having a content defining a plurality of configurations,

characterised in that it has means of controlling the transition from a first state to a second state of the portable electronic object, using means of enabling and/or inhibiting state transitions, so that only some transitions are permitted amongst all the possible transitions.

2. A device according to Claim 1, characterised in that the control means comprise means of checking the content of the volatile memory (3), the data memories (5) and the program memories (4) of the portable electronic object as a function of the state transition permitted to be effected.

3. A device according to either one of Claims 1 or 2, characterised in that the control means enable and/or inhibit a state transition, using a table (11) of permitted state transitions.

4. A device according to Claim 3, characterised in that the control means comprise:

- in addition to the table (11) of permitted state transitions;

- a table (12) of the checks to be made per permitted state transition;

- and a check engine (9) using the said tables.

5. A device according to Claim 3, characterised in that the means of controlling the transition from a first state to a second state of the portable electronic object comprise:

- an extension (16) to the table (11) of permitted state transitions.

6. A device according to Claim 4, characterised in that the means of controlling the transition from a first state to a second state of the portable electronic object comprise:

- an extension (16) to the table (11) of permitted state transitions;

- an extension (17) to the table (12) of checks to be made per permitted state transition;

and in that the check engine (9) uses the said table extensions (16, 17).

7. A device according to any one of Claims 1 to 6, characterised in that the control means comprise means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object.

8. A device according to Claim 7 when the latter depends on Claims 5 or 6, characterised in that the means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object

comprise a table of actions (13) which can be used by the check engine (9).

9. A device according to Claim 8, characterised in that the means for triggering actions during the processing of a request for transition crossover from a first state to a second state of the portable electronic object comprise an extension (18) to the table (13) of actions which can be used by the check engine (9).

10. A portable electronic object having a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), characterised in that it includes the device for controlling the life cycle of the object, according to one of Claims 1 to 9.

11. A smart card having a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), characterised in that it includes the device for controlling the life cycle of the object, according to one of Claims 1 to 9.

12. A method of controlling the life cycle of a portable electronic object, the life cycle being determined by a succession of state transitions, the said states determining the services offered by the object, the said object comprising a processing unit (2), a volatile memory (3), program memories (4) and data memories (5), each of these memories (3, 4, 5) having a content defining a plurality of configurations,

the said method being implemented, within the object, following a state transition request, characterised in that it comprises:

- a step (51, 511, 528, 521) of validation of the enabling of the said request using means of enabling and/or inhibiting state transitions, so that only certain transitions are permitted amongst all the possible transitions;

- a step (57, 517, 527) of modifying the current state of the object if the requested transition is enabled (51, 511, 528, 521).

13. A method according to Claim 12, characterised in that it comprises a step (53, 513, 523) of executing systematic actions, actions associated with the requested transition.

14. A method according to either one of Claims 12 or 13, characterised in that it comprises:

- a step (52, 512, 522) of evaluating checks on the configuration of the object, checks associated with a permitted transition;

- and in that the step (57, 517, 527) of modifying the current state of the object is performed if the said checks on the configuration of the object are satisfied (54, 514, 524).

15. A method according to Claim 14, characterised in that it comprises a step (56, 516, 526) of executing positive actions performed if the requested transition is permitted (51, 511, 528, 521) and if the checks associated with the requested transition are satisfied (54, 514, 524).

16. A method according to either one of Claims 14 or 15, characterised in that it comprises a step

(55, 515, 525) of executing negative actions if the checks associated with the requested transition are not satisfied (54, 514, 524).

17. A method according to either one of Claims 12 to 13, characterised in that it comprises a step (56, 516, 526) of executing positive actions performed if the requested transition is permitted (51, 511, 528, 521).

18. A method according to any one of Claims 12 to 17, implemented within the object, following a request for transition from a first reference state to a second reference state, characterised in that the step (51) of validating the enabling of the said request consists of analysing a table (11) of permitted transitions.

19. A method according to Claim 18, when the latter depends on Claims 13 to 17, characterised in that the step (53) of executing systematic actions consists of:

- using an entry (400, 401), corresponding to the requested transition, in a table (13) of actions, and
- executing a programme of actions (404) defined by the said entry.

20. A method according to either one of Claims 18 or 19, when Claim 18 depends on Claims 14 to 16, characterised in that the step (52) of evaluating the checks associated with the requested transition consists of:

- using an entry (30) in a table (12) of checks, and

- executing a programme (32) of checks defined by the said entry.

21. A method according to any one of Claims 18 to 20, when Claim 18 depends on Claims 15 to 16, characterised in that the step (56) of executing positive actions consists, if the requested transition is enabled (51) and if the checks associated with the requested transition are satisfied (54), of:

- using an entry (400, 402), corresponding to the requested transition, in a table (13) of actions, and
- executing a programme (405) of actions defined by the said entry.

22. A method according to any one of Claims 18 to 21, when Claim 18 depends on Claim 16, characterised in that the step (55) of executing negative actions consists, if the checks associated with the requested transition are not satisfied (54), of:

- using an entry (400, 403), corresponding to the requested transition, in the table (13) of actions, and
- executing a programme (406) of actions defined by the said entry.

23. A method according to either one of Claims 18 to 19, when Claim 18 depends on Claim 17, characterised in that the step (56) of executing positive actions consists, if the requested transition is enabled (51), of:

- using an entry (400, 402), corresponding to the requested transition, in the table (13) of actions, and
- executing a programme (405) of actions defined by the said entry.

24. A method according to any one of Claims 12 to 17, implemented within the object, following a request for transition from a first additive state to a second additive state, characterised in that the step (511) of validating the enabling of the said request consists of analysing an extension (16) to a table (11) of permitted transitions.

25. A method according to Claim 24, when the latter depends on Claims 13 to 17, characterised in that the step (513) of executing systematic actions consists of:

- using an entry (407, 408), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and
- executing a programme of actions (411) defined by the said entry.

26. A method according to either one of Claims 24 or 25, when Claim 24 depends on Claims 14 to 16, characterised in that the step (512) of evaluating the checks associated with the requested transition consists of:

- using an entry (33) in an extension (17) to a table (12) of checks, and
- executing a programme (35) of checks defined by the said entry.

27. A method according to any one of Claims 24 to 26, when Claim 24 depends on Claims 15 to 16, characterised in that the step (516) of executing positive actions consists, if the requested transition

is enabled (511) and if the checks associated with the requested transition are satisfied (514), of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (412) of actions defined by the said entry.

28. A method according to any one of Claims 24 to 27, when Claim 24 depends on Claim 16, characterised in that the step (515) of executing negative actions consists, if the checks associated with the requested transition are not satisfied (514), of:

- using an entry (407, 410), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme (413) of actions defined by the said entry.

29. A method according to either one of Claims 24 to 25, when Claim 24 depends on Claim 17, characterised in that the step (516) of executing positive actions consists, if the requested transition is enabled (511), of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to the table (13) of actions, and

- executing a programme (412) of actions defined by the said entry.

30. A method according to any one of Claims 12 to 17, implemented within the object, following a request for transition from a reference state to an

additive state, characterised in that the step (528, 521) of validating the enabling of the said request consists of:

- validating (528) the enabling of a transition from the said reference state to an additive state, whilst analysing a table (11) of permitted transitions;

- validating (521) the enabling of a transition from the said reference state to the said additive state, whilst analysing an extension (16) to the table (11) of permitted transitions.

31. A method according to Claim 30, when the latter depends on Claims 13 to 17, characterised in that the step (513) of executing systematic actions consists of:

- using an entry (407, 408) corresponding to the requested transition, in an extension (18) to a table (13) of actions, and

- executing a programme of actions (411) defined by the said entry.

32. A method according to either one of Claims 30 or 31, when Claim 30 depends on Claims 14 to 16, characterised in that the step (522) of evaluating the checks associated with the requested transition consists of:

- using an entry (33) in an extension (17) to a table (12) of checks, and

- executing a programme (35) of checks defined by the said entry.

33. A method according to any one of Claims 30 to 32, when Claim 30 depends on Claims 15 to 16,

characterised in that the step (526) of executing positive actions consists, if the requested transition is enabled (528, 521) and if the checks associated with the requested transition are satisfied (524), of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and
- executing a programme (412) of actions defined by the said entry.

34. A method according to any one of Claims 30 to 33, when Claim 30 depends on Claim 16, characterised in that the step (525) of executing negative actions consists, if the checks associated with the requested transition are not satisfied (524), of:

- using an entry (407, 410), corresponding to the requested transition, in an extension (18) to a table (13) of actions, and
- executing a programme (413) of actions defined by the said entry.

35. A method according to either one of Claims 30 to 31, when Claim 30 depends on Claim 17, characterised in that the step (526) of executing positive actions consists, if the requested transition is enabled (528, 521), of:

- using an entry (407, 409), corresponding to the requested transition, in an extension (18) to the table (13) of actions, and
- executing a program (412) of actions defined by the said entry.

36. A method according to any one of Claims 12 to 35, characterised in that the said method does not enable the crossover of a state transition, from an additive state to a reference state.

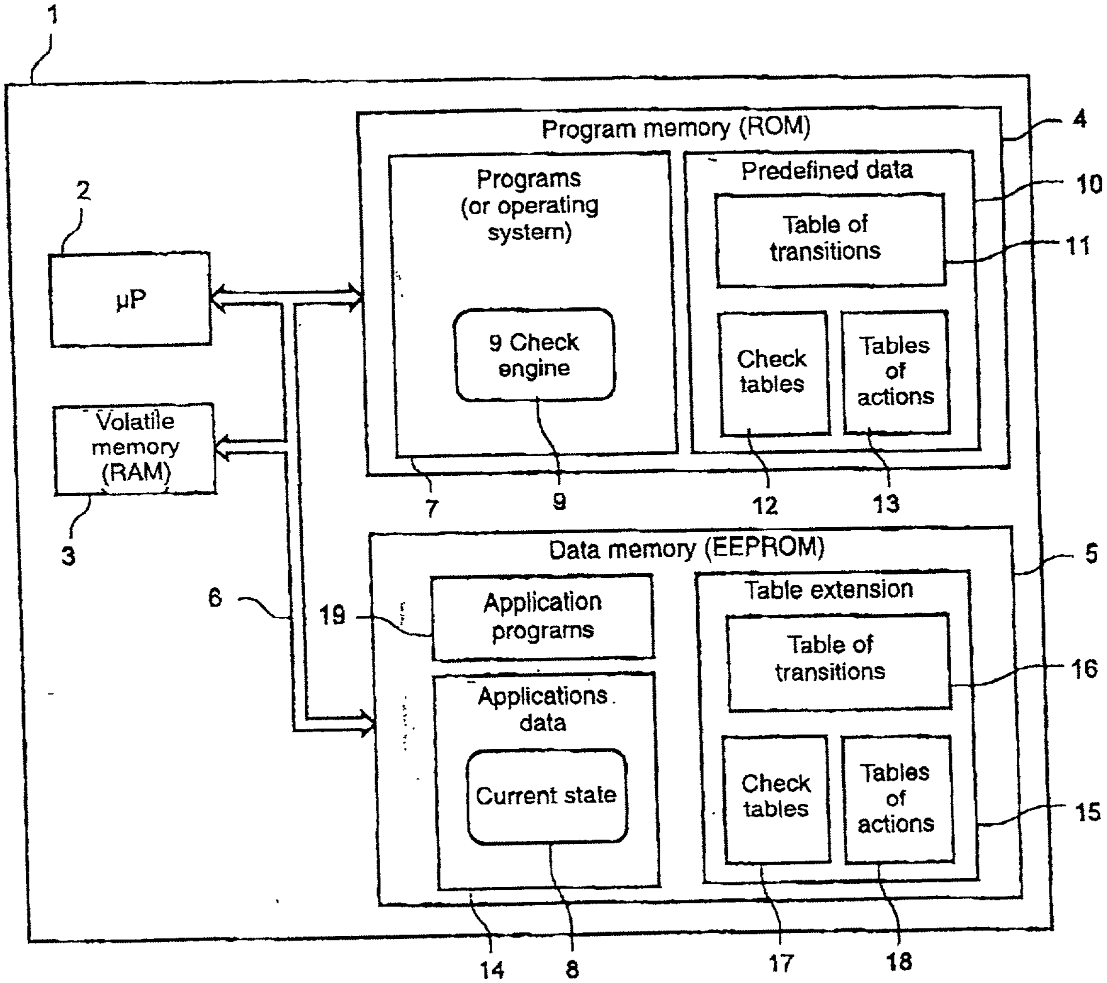

1
Program memory (ROM)
4
Programs
(or operating
system)
Predefined data
10
Table of
transitions
11
2
µP
9 Check
engine
Check
tables
Tables
of
actions
Volatile
memory
(RAM)
7
9
12
13
3
Data memory (EEPROM)
5
6
19
Application
programs
Table extension
Table of
transitions
16
Applications
data
Current state
Check
tables
Tables
of
actions
15
14
8
17
18

FIG. 1

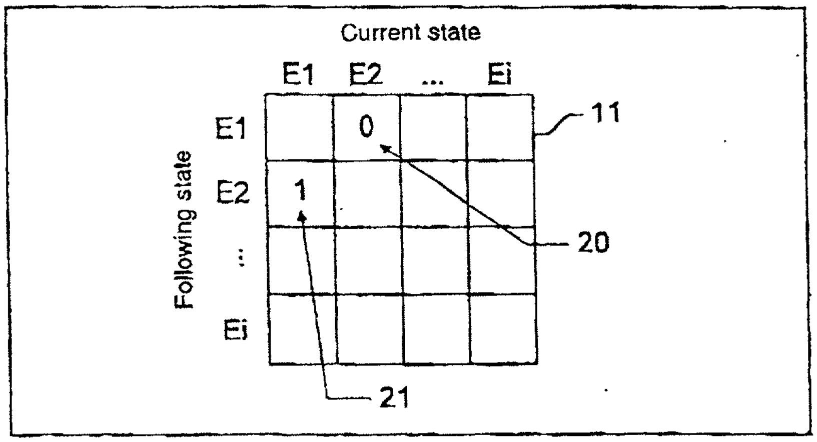

Current state
E1
E2
...
Ei
Following state
E1
E2
⋮
Ei
0
1
11
20
21

FIG. 2a

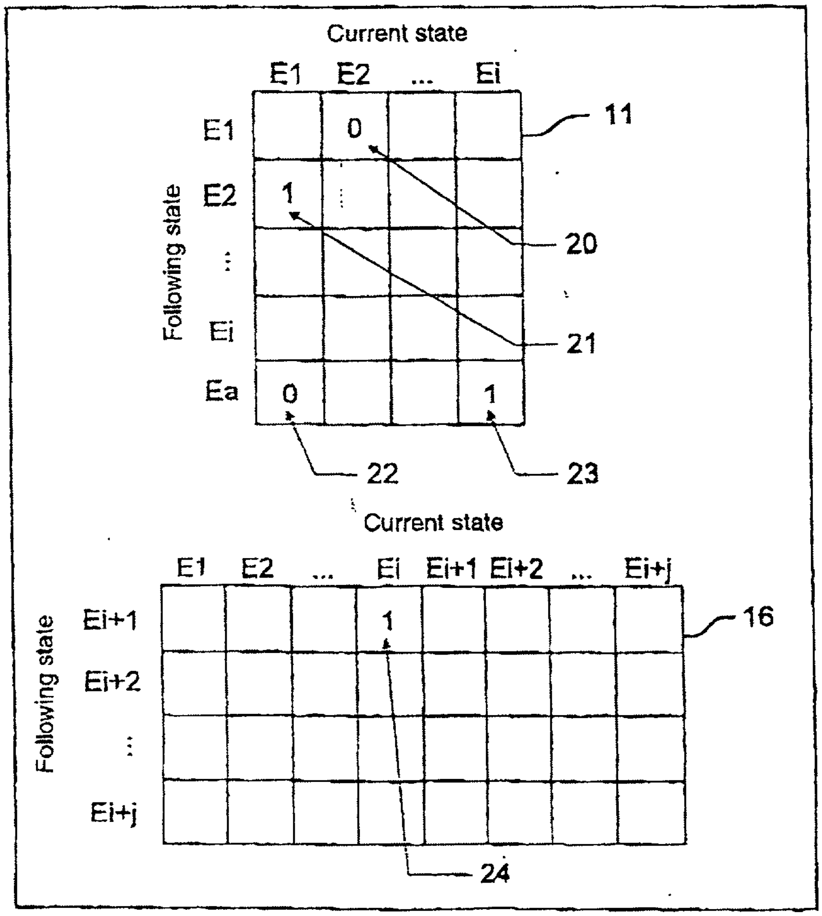

Current state
E1
E2
...
Ei
Following state
E1
E2
⋮
Ei
Ea
0
1
0
1
11
20
21
22
23
Current state
E1
E2
...
Ei
Ei+1
Ei+2
...
Ei+j
Following state
Ei+1
Ei+2
⋮
Ei+j
1
16
24

FIG. 2b

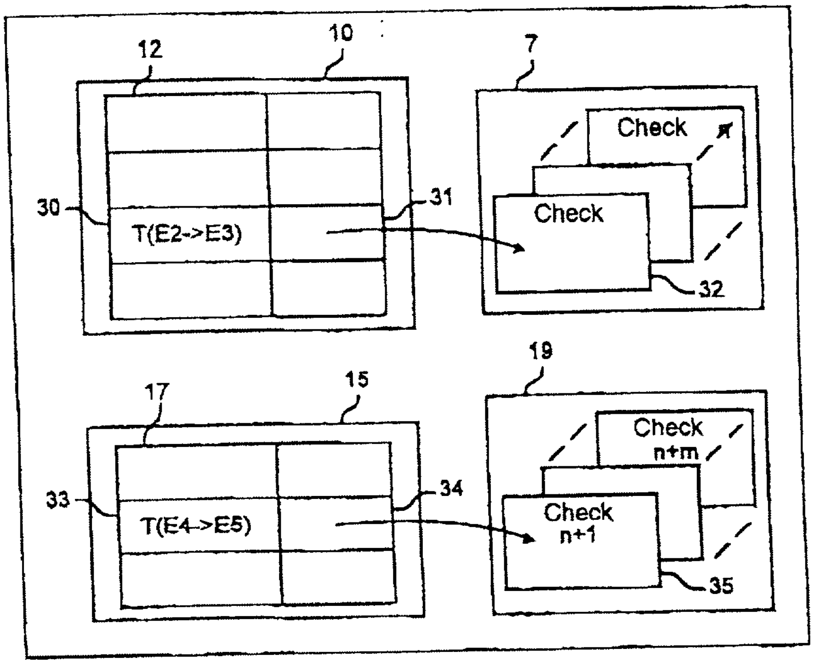

12
10
7
Check
Check
30
31
T(E2->E3)
32
19
17
15
Check
n+m
33
34
Check
n+1
T(E4->E5)
35

FIG. 3

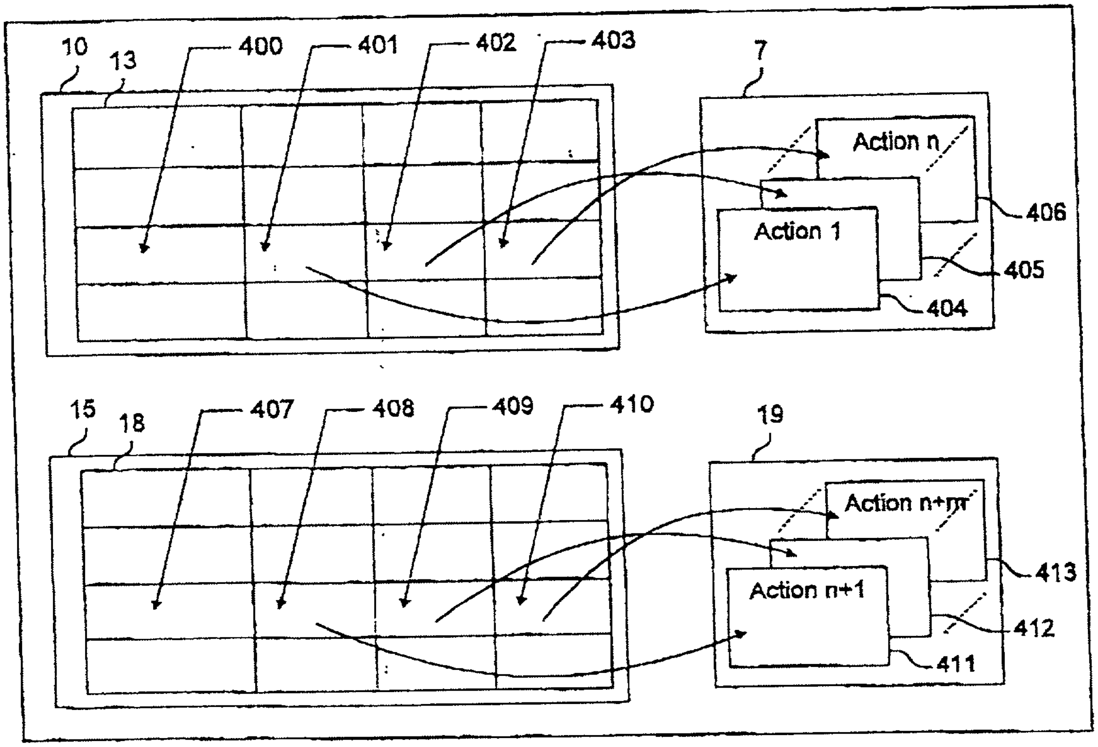

10
13
400
401
402
403
7
Action n
Action 1
406
405
404
15
18
407
408
409
410
19
Action n+m
Action n+1
413
412
411

FIG. 4

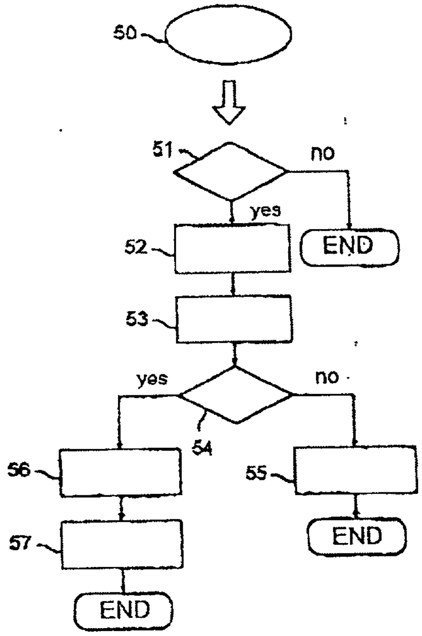


FIG. 5a

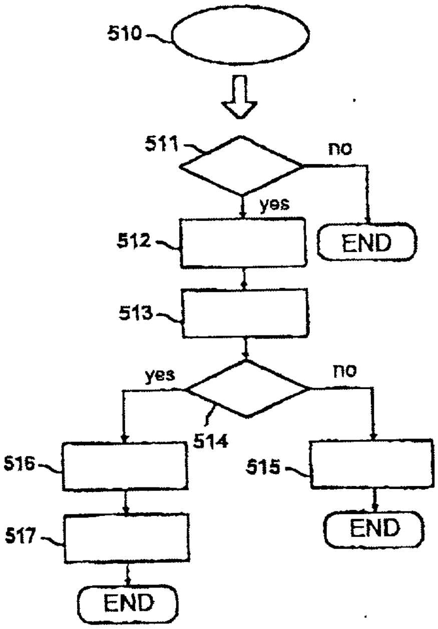


FIG. 5b

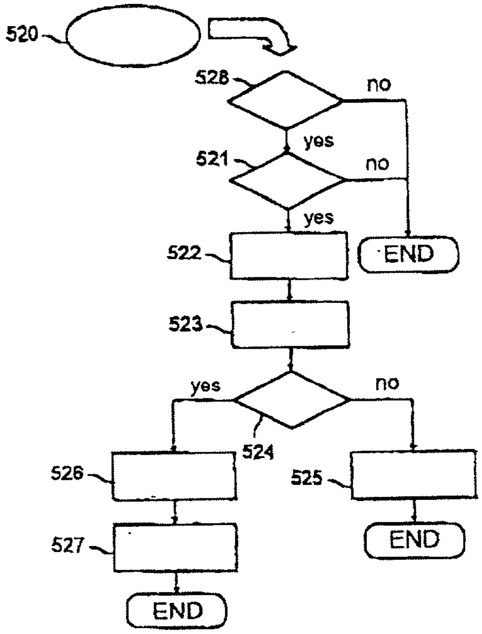


FIG. 5c

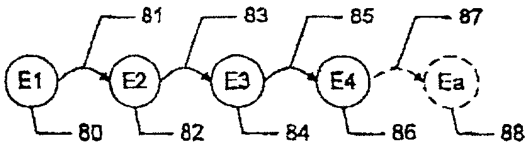
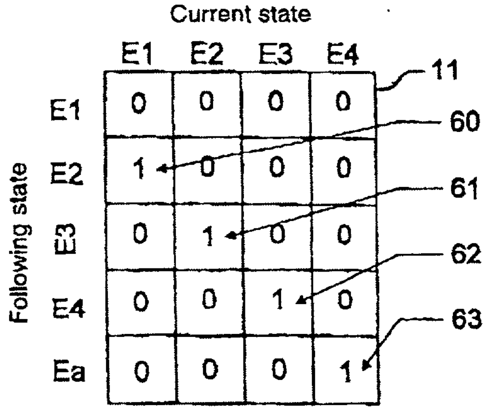


FIG. 6a



	E1	E2	E3	E4
E1	0	0	0	0
E2	1	0	0	0
E3	0	1	0	0
E4	0	0	1	0
Ea	0	0	0	1

FIG. 6b

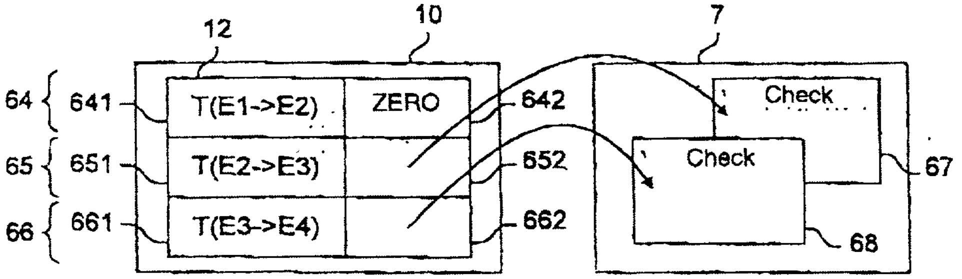


FIG. 6c

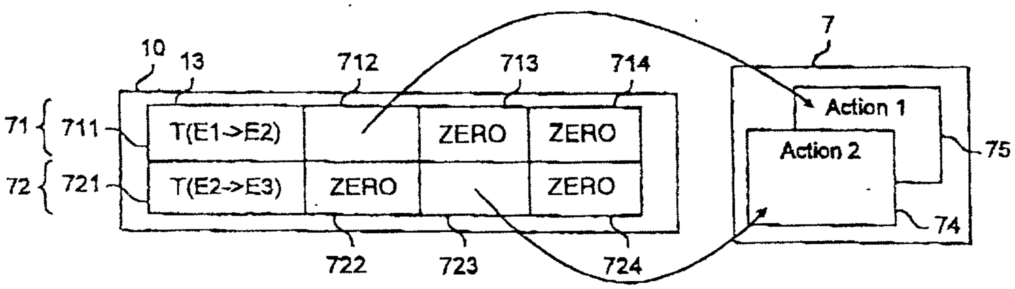


FIG. 6d

COMBINED DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY **(Includes Reference to Provisional and PCT International Applications)**	Attorney's Docket No. 032326-139

As a below named inventor, I hereby declare that:
My residence, post office address and citizenship are as stated below next to my name:
I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND DEVICE FOR CONTROLLING A PORTABLE OBJECT LIFE CYCLE, IN PARTICULAR A SMART CARD

the specification of which (check only one item below):

□ is attached hereto.

☒ was filed as United States application
Number 09/831,745
on May 14, 2001
and was amended
on ______________________ (if applicable).

□ was filed as PCT international application
Number ______________________
on ______________________
and was amended
on ______________________ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 (a)-(e) of any foreign application(s) for patent or inventor's certificate or of any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed:

PRIOR FOREIGN/PCT APPLICATION(S) AND ANY PRIORITY CLAIMS UNDER 35 U.S.C. §119:

COUNTRY (if PCT, indicate "PCT")	APPLICATION NUMBER	DATE OF FILING (day, month, year)	PRIORITY CLAIMED UNDER 35 U.S.C. §119
France	98/14517	13 November 1998	X Yes _No
			_Yes _No
			_Yes _No
			_Yes _No
			_Yes _No

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below.

______________________ (Application Number) ______________________ (Filing Date)

______________________ (Application Number) ______________________ (Filing Date)

COMBINED DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY (CONT'D) (Includes Reference to Provisional and PCT International Applications)	Attorney's Docket No. 032326-139

I hereby claim the benefit under Title 35, United States Code, §120 of any United States applications(s) or PCT international application(s) designating the United States of America that is/are listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in that/those prior application(s) in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose to the Office all information known to me to be material to the patentability as defined in Title 37, Code of Federal Regulations §1.56, which became available between the filing date of the prior application(s) and the national or PCT international filing date of this application:

PRIOR U.S. APPLICATIONS OR PCT INTERNATIONAL APPLICATIONS DESIGNATING THE U.S. FOR BENEFIT UNDER 35 U.S.C. §120:

U.S. APPLICATIONS			STATUS *(check one)*		
U.S. APPLICATION NUMBER	U.S. FILING DATE		PATENTED	PENDING	ABANDONED
PCT APPLICATIONS DESIGNATING THE U.S.					
PCT APPLICATION NO.	PCT FILING DATE	U.S. APPLICATION NUMBERS ASSIGNED (if any)			
PCT/FR99/02678	3 November 1999				

I hereby appoint the following attorneys and agent(s) to prosecute said application and to transact all business in the Patent and Trademark Office connected therewith and to file, prosecute and to transact all business in connection with international applications directed to said invention:

William L. Mathis	17,337	Eric H. Weisblatt	30,505	Bruce T. Wieder	33,815
Robert S. Swecker	19.885	James W. Peterson	26,057	Todd R. Walters	34,040
Platon N. Mandros	22,124	Teresa Stanek Rea	30.427	Ronni S. Jillions	31.979
Benton S. Duffett, Jr.	22.030	Robert E. Krebs	25,885	Harold R. Brown III	36,341
Norman H. Stepno	22,716	William C. Rowland	30.888	Allen R Baum	36,086
Ronald L. Grudziecki	24.970	T. Gene Dillahunty	25.423	Brian P. O'Shaughnessy	32,747
Frederick G. Michaud, Jr.	26,003	Patrick C. Keane	32.858	Kenneth B. Leffler	36,075
Alan E. Kopecki	25.813	B Jefferson Boggs, Jr.	32.344	Fred W. Hathaway	32,236
Regis E. Slutter	26,999	William H Benz	25.952	Wendi L. Weinstein	34,456
Samuel C. Miller, III	27,360	Peter K. Skiff	31.917	Mary Ann Dillahunty	34,576
Robert G. Mukai	28,531	Richard J McGrath	29.195		
George A. Hovanec, Jr.	28.223	Matthew L Schneider	32,814		
James A. LaBarre	28,632	Michael G Savage	32,596		
E. Joseph Gess	28,510	Gerald F Swiss	30.113		
R. Danny Huntington	27.903	Charles F. Wieland III	33,096		

21839

and: ____________________

Address all correspondence to:

21839

James A. LaBarre
BURNS, DOANE, SWECKER & MATHIS, L.L.P.
P O. Box 1404
Alexandria, Virginia 22313-1404

Address all telephone calls to: James A. LaBarre at (703) 836-6620.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

COMBINED DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY (CONT'D) **(Includes Reference to Provisional and PCT International Applications)**	Attorney's Docket No. 032326-139

FULL NAME OF SOLE OR FIRST INVENTOR
Marc BIRKNER
SIGNATURE
DATE 21 08 01
RESIDENCE
2 Résidence Saint Joseph, 13950 Cadolive, FRANCE FRX
CITIZENSHIP
French
POST OFFICE ADDRESS
2 Résidence Saint Joseph, 13950 Cadolive, FRANCE

FULL NAME OF SECOND JOINT INVENTOR, IF ANY
Jean Luc GIRAUD
SIGNATURE
DATE 21/08/01
RESIDENCE
22 rue du Four, 13400 Aubagne, FRANCE FRX
CITIZENSHIP
French
POST OFFICE ADDRESS
22 rue du Four, 13400 Aubagne, FRANCE

FULL NAME OF THIRD JOINT INVENTOR, IF ANY
Laurent TALVARD
SIGNATURE
DATE 29/09/01
RESIDENCE
10 Moutéa Bel Air 13390 Auriol, FRANCE FRX
CITIZENSHIP
French
POST OFFICE ADDRESS
10 Moutéa Bel Air 13390 Auriol, FRANCE

FULL NAME OF FOURTH JOINT INVENTOR, IF ANY
SIGNATURE
DATE
RESIDENCE
CITIZENSHIP
POST OFFICE ADDRESS

FULL NAME OF FIFTH JOINT INVENTOR, IF ANY
SIGNATURE
DATE
RESIDENCE
CITIZENSHIP
POST OFFICE ADDRESS

FULL NAME OF SIXTH JOINT INVENTOR, IF ANY
SIGNATURE
DATE
RESIDENCE
CITIZENSHIP
POST OFFICE ADDRESS

FULL NAME OF SEVENTH JOINT INVENTOR, IF ANY
SIGNATURE
DATE
RESIDENCE
CITIZENSHIP
POST OFFICE ADDRESS

FULL NAME OF EIGHTH JOINT INVENTOR, IF ANY
SIGNATURE
DATE
RESIDENCE
CITIZENSHIP
POST OFFICE ADDRESS